This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1.   (Currently amended)   A method for logging while updating a B-link tree via a plurality of data transactions, comprising:

generating at least one log entry corresponding to a data transaction of the plurality of data transactions, the data transaction to be carried out on said B-link tree, wherein said at least one log entry includes at least one entry from an allocation layer and at least one entry from a B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices; [[and]]

storing said at least one log entry in an intermediate memory previous to storing said at least one log entry in a log; and

storing said at least one log entry into [[a]] the log, wherein said at least one log entry is moved from intermediate memory to the log after the data transaction commits.

2.   (Original)   A method according to claim 1, further including periodically truncating the log.

3.   (Canceled)

4.   (Original)   A method according to claim 1, further including discarding said at least one log entry from the log when the data transaction has been carried out on said B-link tree.

5.   (Original)   A method according to claim 1, wherein said storing includes storing said at least one log entry into the log before the data transaction is carried out on said B-link tree.

6.   (Original)   A method according to claim 1, further including caching data of said data transaction before said data transaction is carried out on said B-link tree.

7 - 8.   (Canceled)

9.   (Original)   A method according to claim 1, further including maintaining a log sequence number with each of said at least one log entry, uniquely identifying said at least one log entry.

10.   (Currently amended)   A ~~computer-readable medium comprising computer executable instructions for performing the~~ method <u>according to</u> ~~of~~ claim 1, wherein said <u>method is embodied in</u> instructions <u>that</u> are stored on ~~said~~ [a] computer readable medium.

11.   (Canceled)

12.   (Currently amended) A method for logging while updating a B-link tree via a plurality of data transactions, comprising:

generating at least one log entry corresponding to a data transaction of the plurality of data transactions, the data transaction to be carried out on said B-link tree, wherein said at least one log entry includes at least one entry from an allocation layer and at least one entry from a B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices;

<u>storing said at least one log entry in an intermediate memory previous to storing said at least one log entry in a finite log;</u>

storing said at least one log entry into [[a]] <u>the</u> finite log<u>, wherein said at least one log entry is moved from intermediate memory to the finite log after the data transaction commits</u>;

periodically flushing data corresponding to data transactions represented by the finite log to persistent storage; and

truncating said finite log in coordination with said flushing.

13.   (Canceled)

14.   (Original)   A method according to claim 12, further including discarding said at least one log entry from the finite log when the data transaction has been carried out on said B-link tree.

15.   (Original)   A method according to claim 12, wherein said storing includes storing said at least one log entry into the finite log before the data transaction is carried out on said B-link tree.

16.   (Original)   A method according to claim 1, further including caching data of said data transaction before said data transaction is carried out on said B-link tree.

17-18. (Canceled)

19.   (Original)   A method according to claim 12, further including maintaining a log sequence number with each of said at least one log entry, uniquely identifying said at least one log entry.

20.   (Currently amended)   A ~~computer readable medium comprising computer executable instructions for performing the~~ method according to ~~of~~ claim 12, wherein said method is embodied in instructions that are stored on ~~said~~ [a] computer readable medium.

21.   (Canceled)

22.   (Currently amended)   A method for logging while updating a data structure via a plurality of data transactions, comprising:

replicating updates to the data structure to a first server location and a second server location;

generating at least one log entry corresponding to a data transaction of the plurality of data transactions, the data transaction to be carried out on said data structure; and

maintaining a single log, where the log is partitioned into an upper layer and an allocation layer, at each of said first and second server locations, wherein the single log includes log entries from both the upper layer and allocation layer, wherein said entries from said allocation layer are local to their corresponding computing devices of a plurality of computing devices and said entries from said ~~B-link tree~~ upper layer are replicated among said plurality of computing devices.

23.   (Original)   A method according to claim 22, further including recovering the data structure after a failure by performing parallel recovery operations by each of said first and second server locations.

24.   (Original)   A method according to claim 22, wherein said data structure is a B-link tree.

25.   (Original)   A method according to claim 24, wherein the upper layer is a B-link tree layer that handles B-link tree operations.

26.   (Original)   A method according to claim 22, wherein the allocater layer handles at least one of (A) an allocate disk space operation, (B) a deallocate disk space operation, (C) a read from the allocated disk space operation and (D) a write to the allocated disk space operation.

27.   (Currently amended)   A ~~computer readable medium comprising computer executable instructions for performing the~~ method according to ~~of~~ claim 22, wherein said method is embodied in instructions that are stored on ~~said~~ [a] computer readable medium.

28.   (Canceled)

29.   (Currently amended)   A server for maintaining a log while updating a B-link tree via a plurality of data transactions, comprising:

a logging object that generates at least one log entry corresponding to a data transaction of the plurality of data transactions, the data transaction to be carried out on said B-link tree;

an allocation layer object for said B-link tree;

a B-link tree layer object, wherein said at least one log entry includes at least one entry from the allocation layer object and at least one entry from the B-link tree layer object, wherein said at least one entry from said allocation layer object is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer object is replicated among said plurality of computing devices; and

a storage log for storing said at least one log entry, including storing said at least one log entry in an intermediate memory previous to storing said at least one log entry in the storage log, wherein said at least one log entry is moved from intermediate memory to the storage log after the data transaction commits.

30.   (Previously Presented)   A server according to claim 29, wherein the storage log including said at least one log entry is periodically truncated.

31.   (Canceled)

32.   (Original)   A server according to claim 29, wherein said at least one log entry is discarded from the storage log when the data transaction has been carried out on said B-link tree.

33.   (Original)   A server according to claim 29, wherein said at least one log entry is stored in the storage log before the data transaction is carried out on said B-link tree.

34.   (Original)   A server according to claim 29, wherein data of said data transaction is cached in a cache memory before said data transaction is carried out on said B-link tree.

35-36. (Canceled)

37.   (Original)   A server according to claim 29, wherein said logging object generates a log sequence number with each of said at least one log entry, uniquely identifying said at least one log entry.

38.   (Currently amended)   A server for logging while updating a B-link tree via a plurality of data transactions, comprising:

        a first object for generating at least one log entry corresponding to a data transaction of the plurality of data transactions, the data transaction to be carried out on said B-link tree;

        an allocation layer and a B-link tree layer, wherein said at least one log entry includes at least one entry from the allocation layer and at least one entry from the B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding

computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices;

a finite storage log for storing said at least one log entry;

a second object for periodically flushing data corresponding to data transactions represented by the at least one log entry in the finite storage log to persistent storage, wherein said persistent storage is configured to receive and store said data after said data transactions commit; and

a third object for truncating said finite storage log in coordination with the operation of the flushing of the second object.

39.   (Canceled)

40.   (Original)   A server according to claim 38, wherein said at least one log entry is discarded from the finite storage log when the data transaction has been carried out on said B-link tree.

41.   (Original)   A server according to claim 38, wherein said at least one log entry is stored in the finite storage log before the data transaction is carried out on said B-link tree.

42.   (Original)   A server according to claim 38, wherein data of said data transaction is cached in a cache memory before said data transaction is carried out on said B-link tree.

43.   (Original)   A server according to claim 38, further including storing said at least one log entry in an intermediate memory previous to storing said at least one log entry in the finite storage log.

44.   (Original)   A server according to claim 43, wherein said at least one log entry is moved from intermediate memory to the finite storage log after the data transaction commits.

45.   (Original)   A server according to claim 38, wherein said first object generates a log sequence number with each of said at least one log entry, uniquely identifying said at least one log entry.

46.   (Previously presented)   A server for logging while updating a data structure via a plurality of data transactions, comprising:

a replication object that replicates updates to the data structure to a first server location and a second server location;

a logging object that generates at least one log entry corresponding to a data transaction of the plurality of data transactions, the data transaction to be carried out on said data structure; and

a storage element within which a single log is maintained, wherein the single log is partitioned into an upper layer and an allocation layer, at each of said first and second server locations, and wherein the single log includes log entries from both the upper layer and allocation layer, wherein said entries from the upper layer are replicated to both said first server and said second server, while said entries from said allocation layer are stored locally on said first server and said second server.

47.   (Original)   A server according to claim 46, wherein the data structure is recovered after a failure via parallel recovery operations by each of said first and second server locations.

48.   (Original)   A server according to claim 46, wherein said data structure is a B-link tree.

49.   (Original)   A server according to claim 48, wherein the upper layer is a B-link tree layer that handles B-link tree operations.

50.   (Original)   A server according to claim 46, wherein the allocater layer handles at least one of (A) an allocate disk space operation, (B) a deallocate disk space operation, (C) a read from the allocated disk space operation and (D) a write to the allocated disk space operation.

51.   (Currently amended)   A computing device for logging while updating a B-link tree via a plurality of data transactions, comprising:

means for generating at least one log entry corresponding to a data transaction of the plurality of data transactions, the data transaction to be carried out on said B-link tree,

wherein said at least one log entry includes at least one entry from an allocation layer and at least one entry from a B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices; and

means for storing said at least one log entry into a log <u>after said data transaction commits</u>.

52.   (Original)   A computing device according to claim 51, further including means for truncating the log periodically.

53.   (Previously presented)   A computing device for logging while updating a B-link tree via a plurality of data transactions, comprising:

means for generating at least one log entry corresponding to a data transaction of the plurality of data transactions, the data transaction to be carried out on said B-tree, wherein said at least one log entry includes at least one entry from an allocation layer and at least one entry from a B-link tree layer, wherein said at least one entry from said allocation layer is local to its corresponding computing device of a plurality of computing devices and said at least one entry from said B-link tree layer is replicated among said plurality of computing devices;

means for storing said at least one log entry into a finite log;

means for periodically flushing data corresponding to data transactions represented by the finite log to persistent storage; and

means for truncating said finite log in coordination with said means for periodically flushing.

54.   (Original)   A computing device according to claim 53, further including means for discarding said at least one log entry from the finite log when the data transaction has been carried out on said B-link tree.

55.   (Previously presented)   A computing device for logging while updating a data structure via a plurality of data transactions, comprising:

means for replicating updates to the data structure to a first server location and a second server location;

means for generating at least one log entry corresponding to a data transaction of the plurality of data transactions, the data transaction to be carried out on said data structure; and

means for maintaining a single log, where the log is partitioned into an upper layer and an allocation layer, at each of said first and second server locations, wherein the single log includes log entries from both the upper layer and allocation layer, wherein said entries from the upper layer are replicated to both said first server and said second server, while said entries from said allocation layer are stored locally on said first server and said second server.

56.   (Original)   A computing device according to claim 55, wherein said data structure is recoverable after a failure by performing parallel recovery operations by each of said first and second server locations.